

Quantum Approximate Optimization Algorithm, introducción, implementación e interpretación.

Marina Maneyro Questa[†], Pablo Martinez Angerosa[‡]

[†]Maestría en Física, Facultad de Ciencias, Udelar

[‡]Licenciatura en Estadística orientación Tecnología, Udelar

Abril 2021

Resumen

En este artículo se presenta el algoritmo heurístico de optimización QAOA. Se introduce en el contexto de los algoritmos variacionales cuánticos, que mezclan la computación clásica y cuántica. Además se detalla su origen basado en la Computación Cuántica Adiabática. En particular, se describe la implementación del algoritmo para el problema del viajero, un problema de implementación muy conocido. Se verifica que el QAOA produce resultados coherentes con las soluciones exactas a los problemas trabajados. Se observa el comportamiento de las distribuciones de probabilidad, y se propone la existencia de convergencias uniformes y su implicancia en el funcionamiento del algoritmo.

1 Introducción

Actualmente la Computación Cuántica se encuentra en una etapa emocionante donde están emergiendo los primeros computadores cuánticos. Esta era de la Computación Cuántica se la conoce como **NISQ**¹ y refleja la situación actual y de futuro cercano de los computadores cuánticos. Los dispositivos se encuentran limitados en la cantidad de qubits que se pueden ejecutar, y por el ruido en las compuertas cuánticas para que de los circuitos creados se obtengan resultados confiables. La corrección de este ruido requeriría un aumento en la cantidad de qubits disponibles en los dispositivos.

Teóricamente las computadoras cuánticas pueden resolver problemas de manera mucho mas eficiente que las computadoras clásicas. Se han presentado algoritmos cuánticos que ampliamente superan a los algoritmos clásicos como el algoritmo de factorización de Shor y el algoritmo de búsqueda de Grover. De momen-

to, estos algoritmos cuánticos se presentan como una opción futura. Se estima que una factorización de interés necesita miles de qubits en un computador cuántico sin corrección de ruido y millones con corrección de ruido, así como un circuito de millones de compuertas. Computaciones de esta escala no son implementables en la era NISQ.

Esta situación requiere algoritmos cuánticos que puedan ser ejecutados en la era NISQ, y aun así resolver problemas útiles. Además, como expone Peter Shor[2], la industria no solo necesita algoritmos cuánticos que representen una ventaja sobre los clásicos, sino que necesita algoritmos que no requieran gran dedicación de expertos para adaptar cada problema a la implementación cuántica.

Es así que en 2014 Edward Farhi, Jeffrey Goldstone y Sam Gutmann [3] presentaron un algoritmo cuántico que produce soluciones aproximadas para problemas de optimización combinatorios. Este algoritmo, llamado **QAOA**² no solo es muy versátil ya que permi-

¹Noisy Intermediate-Scale Quantum (concepto introducido por John Preskill [1])

²Quantum Approximate Optimization Algorithm

te el mapeo de diversos problemas de complejidad NP [4, 5], sino que se puede emplear con alto impacto en su rendimiento computacional en la era de los computadores cuánticos NISQ. Asimismo, se han desarrollado modificaciones a la propuesta original del QAOA, que podrían permitir aún mayor efectividad. Por esta gran utilidad, el QAOA se ha convertido en uno de los algoritmos cuánticos mas relevantes de la actualidad.

Este artículo se organiza en cinco secciones. En la Sección 2 se describe el algoritmo QAOA, partiendo del concepto de algoritmo heurístico computacional y relacionándolo con la Computación Adiabática. Además introducimos el problema del viajero, al que vamos a aplicar el algoritmo. En la Sección 3 se describe la implementación del QAOA en el circuito cuántico así como los medios tecnológicos donde fue ejecutado para el desarrollo de este artículo. En la Sección 4 se muestran los resultados de la implementación y se presenta la posibilidad de la existencia de convergencias uniformes que determinan una interpretación del funcionamiento del QAOA. En la Sección 5 se presentan las conclusiones y pasos futuros relacionados a investigar.

2 Marco Teórico

En esta sección se introduce el algoritmo QAOA, su origen, y algunos detalles sobre su utilización. El QAOA es un **algoritmo variacional** inspirado en la **Computación Adiabática**. Se utiliza principalmente en problemas de optimización combinatoria, aunque posee una gran flexibilidad. En años recientes se ha probado la universalidad del mismo e incluso fue propuesto como candidato para demostrar la supremacía cuántica³.

Este algoritmo es de baja profundidad (requiere un circuito de relativamente pocas puertas), lo que significa que no requiere demasiada coherencia. Algunos estudios demuestran que sus resultados son robustos da-

do el error existente en los computadores cuánticos actuales.

Los qubits que el QAOA necesita para su ejecución son solo los requeridos para almacenar la información cuántica del problema buscado. Esto lo hace inviable para la solución de problemas de una escala gigante en la era NISQ. Aun así, para la cantidad de qubits disponible, el algoritmo iguala el poder de cómputo de los computadores clásicos actuales mas potentes.

2.1 Algoritmo QAOA

2.1.1. QAOA como algoritmo heurístico

Se pueden distinguir dos clases de algoritmos, los que tienen un rendimiento demostrado y los algoritmos heurísticos. Los algoritmos heurísticos se implementan cuando, debido a la dificultad de un problema, se descarta la posibilidad de obtener soluciones exactas. Como alternativa, se buscan atajos para llegar a soluciones satisfactorias aceptando pérdidas de precisión. Dentro de los algoritmos exactos con rendimiento demostrado se puede encontrar a modo de ejemplo la multiplicación de matrices (clásico) y los algoritmos de Shor y Grover (cuánticos). Por otro lado, dentro de los algoritmos heurísticos mas utilizados en un enfoque clásico encontramos el descenso por gradiente, el *annealing* simulado y los algoritmos genéticos. En la perspectiva cuántica encontramos la clase de Algoritmos Variacionales, que se desarrolla en la subsección 2.1.2.

En la actualidad los algoritmos heurísticos clásicos son los mas utilizados y con resultados tecnológicos mas disruptivos. Es de esperar que dentro de una perspectiva cuántica, los algoritmos heurísticos cumplan el mismo rol.

2.1.2. Algoritmos Variacionales Cuánticos y circuitos *Ansatz*

El QAOA pertenece a la familia de Algoritmos Variacionales Cuánticos. Estos algoritmos tienen una estructura inspirada en las redes neuronales y su arquitectura de capas, en

³La supremacía cuántica, alternativamente llamada ventaja cuántica, es la capacidad potencial de los dispositivos de computación cuántica para resolver problemas que los ordenadores clásicos prácticamente no pueden resolver [6]

las cuales se da un proceso iterativo de optimización de parámetros basados en una función de costo[7].

Los algoritmos variacionales parten de un estado *Ansatz*, que depende de ciertos parámetros, para los que se buscarán los valores que minimicen la función costo.⁴

Un circuito de profundidad p es una secuencia de compuertas que se puede escribir como $\Pi_{l=1}^p U_l |0\rangle^{\otimes n}$. La profundidad indica a cantidad de operaciones U que conforman el circuito *Ansatz*.

Un estado *Ansatz* dado $|\psi(\theta)\rangle$ (parame-trizado por θ), genera un espacio de estados posibles Γ que se denomina **Espacio de Estados Variacionales**. El espacio de estados depende de la estructura y profundidad del *Ansatz*. En la Ec. (1) se define el espacio de estados variacionales como la unión de todos los estados posibles de un *Ansatz*, para todos los posibles valores de los ángulos $\theta \in (0, 2\pi]^{\times p}$

$$\Gamma = \bigcup_{\theta \in (0, 2\pi]^{\times p}} |\psi(\theta)\rangle \quad (1)$$

Es importante destacar que para la mayoría de los *Ansatz* el espacio generado Γ es desconocido, y no es necesariamente el mismo espacio de Hilbert que lo contiene, ni un espacio vectorial. Asimismo, la solución exacta del problema a resolver no necesariamente pertenece al espacio Γ , debido a la aproximación característica de los algoritmos variacionales basados en *Ansatz*.

En el Algoritmo(1) se muestra el esquema completo de un algoritmo cuántico variacional. La estructura es sencilla. Primariamente se inicializa el estado del circuito *Ansatz* mediante una lista de parámetros reales θ . Se procede a ejecutar valores esperados mediante un Hamiltoniano H_C que define el problema. Mediante una función de costo se computan los valores esperados y se procede, mediante una rutina de optimización clásica, a fijar los nuevos valores de θ para repetir el procedimiento. De este modo se genera un bucle de optimización híbrido entre el mundo clásico

y cuántico. Esta interacción entre el cómputo clásico y el cómputo cuántico, es una característica esencial de los algoritmos variacionales cuánticos.

Algoritmo 1 Algoritmo Variacional Cuántico

- 1: $\theta \leftarrow \text{init}$ /* estado inicial */
 - 2: **while do**
 - 3: **Estados:** una lista de parámetros reales θ configuran el circuito que genera el estado $|\theta\rangle$
 - 4: **Mediciones:** para obtener el valor esperado de un Hamiltoniano $\langle\psi(\theta)|H_C|\psi(\theta)\rangle$
 - 5: **Función de costo** clásica se computa para cada $|\theta\rangle$.
 - 6: **Bucle-externo optimización** rutina de optimización clásica fija los nuevos parámetros $\theta \Rightarrow \theta_\star$
 - 7: **end while**
-

2.1.3. Hamiltoniano Cuántico

En las subsecciones 2.1.1 y 2.1.2, se explicó que el QAOA es un algoritmo heurístico, que pertenece a la familia de los algoritmos variacionales cuánticos. Para comprender la implementación de este tipo de algoritmos debemos conocer las características de los operadores Hamiltonianos.

Un Hamiltoniano es un operador hermítico y por ende podemos considerarlo simplemente como una matriz con ciertas propiedades y restricciones. Para los algoritmos variacionales relacionaremos a los Hamiltonianos con la función de costo booleana $f(x)$ del problema a solucionar, para $x \in \{0, 1\}^n$. En particular, nos interesa la clase de Hamiltonianos definidos mediante una matriz diagonal $2^n \times 2^n$, donde cada elemento de la diagonal contiene una posible evaluación f [4]. En la Ec. (2) se muestra la matriz que representa a un Hamiltoniano H_C con la estructura buscada, donde nuestro objetivo es encontrar $\arg \min_{x \in \{0, 1\}^n} f(x)$

⁴Se entiende por estado *Ansatz* a un estado de prueba cuántico, que se considera lo suficientemente aproximado al estado objetivo preciso que se desea (una vez que se realice la optimización).

$$\begin{pmatrix} f(0, \dots, 0) & 0 & \dots & 0 \\ 0 & f(0, \dots, 1) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & f(1, \dots, 1) \end{pmatrix} \quad (2)$$

Con este Hamiltoniano se puede efectuar $H_C |x\rangle = f(x) |x\rangle \forall x \in \{0, 1\}^n$. Se define como **estado base (ground state)** al estado asociado a la energía mínima del Hamiltoniano ($|x_0\rangle : f(x_0) = \min f(x)$).

En la Ec. (3) se reescribe el Hamiltoniano en términos de los proyectores Pr_x :

$$H_C = \sum_{x \in \{0, 1\}^n} f(x) |x\rangle \langle x| = \sum_{x \in \{0, 1\}^n} f(x) Pr_x \quad (3)$$

Aplicando la Ec. (3) se puede definir el valor esperado del Hamiltoniano para un estado *Ansatz* como:

$$\begin{aligned} & \langle \psi(\theta) | H | \psi(\theta) \rangle \\ &= \langle \psi(\theta) | \sum_{x \in \{0, 1\}^n} f(x) Pr_x | \psi(\theta) \rangle \\ &= \sum_{x \in \{0, 1\}^n} f(x) \langle \psi(\theta) | Pr_x | \psi(\theta) \rangle \\ &= \sum_{x \in \{0, 1\}^n} f(x) p(x) \end{aligned}$$

Estos Hamiltonianos son demasiado grandes para construirlos de un modo explícito, por lo que se utilizan formulaciones de construcción compactas y estandarizadas basadas en matrices de Pauli Z (σ_z). En la sección 2.4 se detalla la formulación en un circuito cuántico de un Hamiltoniano enfocado específicamente al problema del viajero.

2.1.4. Algoritmo QAOA

En las subsecciones previas se mostró como el QAOA tiene una estructura de preparación de un estado cuántico *Ansatz*. El QAOA prepara el estado *Ansatz* de forma inspirada en la Computación Cuántica Adiabática, como se detalla en la sección 2.2. El estado es parametrizado con dos listas reales γ y β .

Dado un Hamiltoniano H_C dado por la Ec. (3), su valor esperado para el estado *Ansatz* se relaciona con el mínimo de la función de costo de la siguiente manera:

$$\min_{z \in \{0, 1\}^n} f(z) = \min_{|\psi\rangle \in \mathbb{C}_2^{\otimes n}} \langle \psi | H_C | \psi \rangle \quad (4)$$

Se define el espacio variacional de estados Ω de un QAOA de profundidad p como:

$$\Omega = \bigcup_{\gamma, \beta} \{ |\psi(\gamma, \beta)\rangle \} \subseteq \mathbb{C}_2^{\otimes n}$$

donde se cumple que:

$$\begin{aligned} |\psi(\theta)\rangle &= |\psi(\gamma, \beta)\rangle \\ &= \prod_{k=1}^p e^{-i\beta_k B} e^{-i\gamma_k H_C} (H|0\rangle)^{\otimes n} \\ &= e^{-i\beta_p B} e^{-i\gamma_p H_C} \dots e^{-i\beta_1 B} e^{-i\gamma_1 H_C} (H|0\rangle)^{\otimes n} \end{aligned} \quad (5)$$

Donde B es el Hamiltoniano *Mixer* usualmente definido por $B = \sum_i X_i$ y H representa a la puerta Hadamard. El estado de superposición inicial $(H|0\rangle)^{\otimes n} = |+\rangle^{\otimes n}$ es un autovector del *Mixer* B , y representa la distribución donde todos los estados del sistema son equiprobables. La idea detrás algoritmo es evolucionar este estado hasta obtener una distribución donde el estado base del Hamiltoniano H_C se obtiene con mayor probabilidad [8].

La aproximación del algoritmo queda determinada por

$$\min_{\psi \in \Omega} \langle \psi | H_C | \psi \rangle \geq \min_{\phi \in \mathbb{C}_2^{\otimes n}} \langle \phi | H_C | \phi \rangle$$

Siguiendo la estructura del Algoritmo 1, se utiliza un optimizador clásico para variar los parámetros γ y β y recorrer en una forma óptima el espacio Ω generado por el *Ansatz*, en búsqueda de minimizar el valor esperado dado por $f(\beta, \gamma) = \langle \psi(\beta, \gamma) | H_C | \psi(\beta, \gamma) \rangle$. Se encuentran los parámetros γ^* β^* que cumplan

$$\langle \psi(\gamma^*, \beta^*) | H_C | \psi(\gamma^*, \beta^*) \rangle \approx \min(H_C)$$

En el artículo original[3] de QAOA, se plantea que, llamando m_p al valor esperado obtenido al hallar parámetros para una profundidad dada, el comportamiento observado es

$m_p \leq m_{p-1}$. Además se demuestra como un $p \rightarrow \infty$ lleva a un resultado óptimo, como se puede ver en la sección 2.2. Sin embargo para p mas chicos, la práctica muestra resultados aproximados satisfactorios del algoritmo, como se puede ver en la implementación de la sección 4.

2.2 QAOA y Computación Cuántica Adiabática

El algoritmo QAOA fue inspirado por un método de computación cuántica que no se basa en puertas. Esta alternativa para resolver problemas computacionales utilizando sistemas cuánticos es conocida como Computación Cuántica Adiabática. A continuación se detalla el proceso de *Annealing* Cuántico y el Teorema Adiabático, fundamentos de la Computación Adiabática. Luego se especifica su relación con los circuitos de puertas cuánticas con los que se implementa el QAOA.

2.2.1. *Annealing* Cuántico y Computación Cuántica Adiabática

La Computación Adiabática es un enfoque de computación cuántica, *a priori* equivalente a las computadoras de puertas [9]. Sin embargo, los modos de operar sobre los qubits se basan en paradigmas muy diferentes. Debido a sus mecanismos de funcionamiento, las aplicaciones de Computación Adiabática más desarrolladas se han centrado en problemas de optimización. A medida que la comprensión de este enfoque ha evolucionado, se ha posicionado como capaz de solucionar problemas más generales, al igual que la computación por puertas[10].

El *Annealing* Cuántico es un proceso que permite interpolar entre un operador Hamiltoniano sencillo, con autoestados conocidos, y otro (más complejo) que representa el problema que se desea solucionar. Esto hace que sea útil para resolver problemas de optimización, donde se busca hallar el estado fundamental del Hamiltoniano del problema, llevando el sistema a una configuración donde este estado sea medido con mayor probabilidad.

En el caso de la Computación Adiabática,

se busca que la evolución del sistema cuántico entre los Hamiltonianos tenga características particulares. De esta forma, el proceso se regirá por el Teorema Adiabático. En el contexto de la Computación Cuántica, un proceso adiabático se define de forma diferente a la Termodinámica Clásica. Se debe trabajar con evoluciones cuasi-estáticas, lo suficientemente lentas como para que el sistema no experimente saltos en el nivel de energía. El Teorema Adiabático indica que un sistema cuántico se mantendrá en un mismo autoestado instantáneo durante su evolución si se cumplen dos condiciones. Las perturbaciones a las que se ve sujeto deben actuar lentamente, y debe existir una separación entre este estado y los otros estados del espectro Hamiltoniano. Durante este tipo de evolución el sistema puede adaptar su configuración de forma que, si se parte de un estado particular del Hamiltoniano sencillo, se arribe en el estado correspondiente para el Hamiltoniano del problema.

Un algoritmo de Computación Adiabática puede ser descrito mediante un Hamiltoniano de la siguiente forma:

$$\hat{H}(t) = (1 - s(t))\hat{H}_M + s(t)\hat{H}_C, \quad (6)$$

donde \hat{H}_M es el Hamiltoniano sencillo del que se parte (comúnmente llamado *driver* o *mixer*) y \hat{H}_C describe el problema (Hamiltoniano Costo). La función suave $s(t)$ representa la evolución temporal. Debido a que el estado final estará dado por \hat{H}_C , y el inicial por \hat{H}_M , $s(t)$ debe cumplir $s(0) = 0$ y $s(T) = 1$ (T corresponde al tiempo final). El comportamiento de $s(t)$, y el valor de T se deben elegir de acuerdo a las condiciones que requiere el Teorema Adiabático.

Físicamente, los algoritmos de este tipo se pueden implementar mediante sistemas de *spins*, que evolucionan mediante la variación de un campo magnético externo. Esta es la base del funcionamiento de computadoras como las de la empresa D-wave. Este tipo de computadoras es útil para solucionar problemas que se pueden escribir con la misma forma que un Hamiltoniano del modelo Ising. Estos Hamiltonianos describen la energía de conjuntos de *spins* interactuantes en la presencia de un

campo magnético, que en las computadoras suele ser regulable.

Si se desea implementar este procedimiento en computadoras basadas en puertas lógicas, se debe escribir en términos de operadores de Pauli no superpuestos (que actúan sobre qubits individuales). Entonces, se realiza una descomposición aproximada de la evolución temporal en pasos discretos y se separa la contribución de cada Hamiltoniano (Costo y *mixer*) en operaciones de puertas. La técnica es conocida como *Trotterización*.

2.2.2. Trotterización

Podemos definir un operador unitario que representa la evolución temporal del sistema cuántico bajo el Hamiltoniano de ecuación (6).

$$U(T, 0) = \tau \exp \left[-i \int_0^T H(t) dt \right] \quad (7)$$

Este representa la evolución entre el tiempo 0 y el tiempo final T , definida de la forma usual a partir del operador de tiempo ordenado τ .

Es posible separar la evolución en pasos de duración $\Delta t = T/N$, mediante un producto de N operadores de evolución:

$$\begin{aligned} U(T, 0) &= \\ U(T, T - \Delta t) U(T - \Delta t, T - 2\Delta t) \dots U(\Delta t, 0) \\ &= \prod_{k=1}^N U(k\Delta t, (k-1)\Delta t) \end{aligned} \quad (8)$$

Si $\Delta t \ll T$ se puede considerar que el Hamiltoniano es constante en cada uno de esos pasos. Entonces se aproxima el operador evolución como:

$$U(T, 0) \approx \prod_{k=1}^N e^{-i\Delta t H(k\Delta t)}. \quad (9)$$

Utilizando la fórmula de Trotter-Suzuki ($e^{\delta(A+B)} \approx e^{\delta A} e^{\delta B} + \mathcal{O}(\delta^2)$, para $\delta \sim 0$), podemos separar los exponenciales correspondien-

tes a los dos Hamiltonianos.

$$\begin{aligned} U(T, 0) &\approx \prod_{k=1}^N e^{-i\Delta t[(1-s(k\Delta t))H_A + s(k\Delta t)H_B]} \\ &\approx \prod_{k=1}^N e^{-i\Delta t(1-s(k\Delta t))H_A} e^{-i\Delta t s(k\Delta t)H_B} \end{aligned} \quad (10)$$

Realizando algunas sustituciones, es posible recuperar las rotaciones que según el algoritmo QAOA nos permiten obtener los estados $|\vec{\gamma}, \vec{\beta}\rangle$. En lugar de N , se consideran p pasos, y por lo tanto p evoluciones según cada Hamiltoniano. Además se definen $2p$ parámetros libres $\beta_k \rightarrow \Delta t(1 - s(k\Delta t))$ y $\gamma_k \rightarrow \Delta t s(k\Delta t)$ [11]. De este modo se obtiene el operador:

$$Q_p = \prod_{k=1}^p e^{-i\gamma_k H_A} e^{-i\beta_k H_B}. \quad (11)$$

Al implementar el QAOA tendremos $p \ll N$ debido a que aumentar p requiere un circuito cuántico con mayor profundidad, que introducirá error respecto a un proceso adiabático ideal. Además no obtendremos los factores β_k y γ_k adecuados a partir de la elección de $s(t)$ y Δt , sino utilizando métodos de optimización clásicos (dado un p fijo).

El QAOA se también diferencia de las implementaciones del *annealing* y la computación adiabática existentes de otras maneras. Por ejemplo, al buscar la evolución adecuada en una computadora adiabática, aumentar el tiempo de evolución no garantiza mejoras monotónicas de los resultados. Sin embargo, en el QAOA se garantiza mejores resultados para mayor p . Además puede ocurrir que el QAOA proporcione el mínimo global en problemas para los que la Computación Adiabática quede estancada en un mínimo local.

2.3 Problemas QUBO y modelo Ising

El QAOA y el *Annealing* cuántico suelen ser aplicados para problemas descritos por Hamiltonianos y funciones de costo de forma particular. Estos modelos se suelen categorizarse como modelos cuadráticos binarios. Las formulaciones de este tipo con los que se trabaja son el modelo Ising y los problemas QUBO.

El modelo Ising proviene de la mecánica estadística, pero suele ser útil para la descripción de una gama amplia de comportamientos. Originalmente describe sistemas compuestos por espines *up* ($\sigma_i = +1$) y *down* ($\sigma_i = -1$). Los espines pueden interactuar entre sí y con un campo magnético externo. El Hamiltoniano correspondiente a este problema tiene la siguiente forma:

$$H(\sigma) = \sum_{i=1}^N h_i \sigma_i + \sum_{i=1}^N \sum_{j=i+1}^N J_{i,j} \sigma_i \sigma_j. \quad (12)$$

Los factores h_i representan la interacción del espín i con el campo externo, y los coeficientes $J_{i,j}$ la interacción entre los espines i y j .

Los problemas QUBO (*Quadratic Unconstrained Binary Optimization*) tienen variables $x_i = \{0, 1\}$ (verdadero o falso). Las funciones objetivo para este tipo de problemas son

$$f(x) = \sum_{i=1}^N q_{ii} x_i + \sum_{i < j} q_{ij} x_i x_j. \quad (13)$$

Es posible probar que los problemas Ising y QUBO son equivalentes definiendo variables $\{0, 1\}$ a partir de los espines $\sigma_i = \{1, -1\}$. [12]

Los problemas de optimización en Vidrios de espín de Ising (hallar estado de menor energía) son NP-Hard, y no se ha logrado diseñar algoritmos exactos que los solucionen en tiempo polinomial [5]. La dificultad para obtener resultados para estos casos, como en el problema del viajero que se discutirá a continuación, ha fomentado interés en el desarrollo de soluciones heurísticas clásicas y cuánticas.

2.4 El problema del viajero

El Problema del Viajero (*Travelling Salesman Problem*) es un problema de optimización muy estudiado [5]. Consiste en hallar el camino mas corto en el que se puede recorrer un conjunto de ciudades, dadas las distancias entre ellas. El problema es difícil de solucionar computacionalmente, ya que la cantidad de caminos posibles aumenta exponencialmente con la cantidad de ciudades a recorrer. En consecuencia, el tiempo requerido para obtener la

solución exacta por fuerza bruta (comparando las distancias recorridas en cada camino) se vuelve impráctico aún con pocas ciudades. Por eso resulta de interés aplicar el algoritmo QAOA a este problema en particular.

Como se detalló previamente, para implementar el QAOA para este problema, debemos expresarlo como un Hamiltoniano de Ising. Por eso, definimos una variable binaria $x_{i,l}$ que indica si el viajero está en la ciudad i en tiempo l (1 en caso afirmativo y 0 si se encuentra en otro lado). A la distancia entre las ciudades i y j le llamamos $J_{i,j}$, y al conjunto de los caminos que unen las ciudades es E .

La distancia recorrida en N intervalos de tiempo (N ciudades) está dada por la siguiente función costo:

$$f_{Ising} = \sum_{i,j \in E} J_{i,j} \sum_l x_{i,l} x_{j,l+1}. \quad (14)$$

El término $x_{i,l} x_{j,l+1}$ es igual a 1 cuando el viajero esta en i en el tiempo l y además esta en j en el tiempo siguiente ($l+1$), por lo que si una de estas condiciones no se cumple no se suma ninguna distancia.

También debemos tener en cuenta las restricciones que se prefiere que se cumplan, o que deben cumplirse por limitaciones de la realidad. Por ejemplo, el viajero no debería pasar mas de un intervalo de tiempo en el mismo lugar o pasar mas de una vez por la misma ciudad. Entonces incluimos otra función que penalice estos casos, o sea, aumente el costo lo suficiente como para descartar como mínimos los caminos que no cumplan estas condiciones:

$$f_{constr} = \lambda \left[\sum_{l=1}^N \left(1 - \sum_{i=1}^N x_{i,l} \right)^2 + \sum_{i=1}^N \left(1 - \sum_{l=1}^N x_{i,l} \right)^2 + \sum_{i,j \notin E} \sum_{l=1}^N x_{i,l} x_{j,l+1} \right] \quad (15)$$

El primer término penaliza los casos donde está en mas de una ciudad en el mismo tiempo, el segundo los casos donde pasa mas de un tiempo en la misma ciudad, y el tercero penaliza los casos donde se mueve por conexiones que no existen. Cuando el camino del viajero

satisface las condiciones deseadas f_{constr} se minimiza a cero. La variable λ (multiplicador de Lagrange) se ajusta de forma que las penalizaciones sean lo suficientemente grandes para producir buenos resultados, por ejemplo tomando $\lambda > \max J_{i,j}$.

Para escribir el Hamiltoniano en términos de operadores unitarios, utilizamos matrices de Pauli $\sigma_{i,l}^z$ con autovalores 1 y -1, y reescribimos las variables $x_{i,l}$ de forma que aplicar el Hamiltoniano a un estado cuántico sea equivalente a evaluar la función $f(x)$, como en la sección 2.1.3.

$$x_{i,l} \rightarrow \frac{1}{2} (1 + \sigma_{i,l}^z) = \begin{cases} 1 & \text{si } \sigma_{i,l}^z = 1 \\ 0 & \text{si } \sigma_{i,l}^z = -1 \end{cases} \quad (16)$$

Los coeficientes de las matrices $\sigma_{i,l}^z$ indican que se corresponderán a rotaciones en torno al eje z, y que cada una de ellas actúa sobre un qubit distinto (i, l) . Entonces el Hamiltoniano que queremos solucionar tendrá autoestados $|x_{0,0}, x_{1,0}, x_{2,0}, \dots, x_{N,N}\rangle$. Estará compuesto por la suma de f_{Ising} y f_{constr} escritos en términos de los $\sigma_{i,l}^z$. De esta forma el Hamiltoniano resultante está formado por términos que multiplican a una o dos matrices de Pauli Z, a los que nos referiremos como Z o ZZ.

3 Implementación

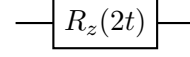
3.1 Circuito Cuántico

En la subsección 2.1.4 se mostró que el *Ansatz* del QAOA de profundidad p queda determinado por la ecuación (5). Para implementarlo en un circuito cuántico es necesario hacer algunas transformaciones. Primariamente es necesario entender cómo implementar los términos exponenciales del Hamiltoniano H_C en el circuito, a partir de los elementos Z y ZZ que lo componen [13], como se detalla al final de la sección 2.4.

El elemento mas simple es el operador e^{-iZt} que se puede recrear en el circuito con la rotación en torno al eje z $R_z(2t)$, ya que

$$R_z(\theta) \equiv e^{-i\theta Z/2} \implies e^{-iZt} \equiv R_z(2t)$$

En el circuito el operador queda implementado como:



Un operador mas complicado que aparece en la ecuación es

$$e^{-iZZt} = e^{-iZ \otimes Zt}$$

Para resolver la implementación en el circuito de este operador es necesario recordar la propiedad

$$e^A |v\rangle = e^\lambda |v\rangle$$

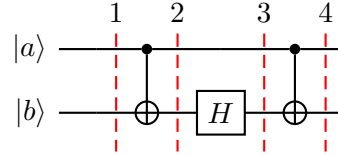
si $A|v\rangle = \lambda|v\rangle$ y que Z tiene autovalores 1, -1 y autovectores $|0\rangle$ y $|1\rangle$. Entonces se tiene

$$\begin{aligned} e^{-iZ \otimes Zt} |00\rangle &= e^{-i(1 \times 1)t} |00\rangle = e^{-it} |00\rangle \\ e^{-iZ \otimes Zt} |01\rangle &= e^{-i(1 \times -1)t} |01\rangle = e^{it} |01\rangle \\ e^{-iZ \otimes Zt} |10\rangle &= e^{-i(-1 \times 1)t} |10\rangle = e^{it} |10\rangle \\ e^{-iZ \otimes Zt} |11\rangle &= e^{-i(-1 \times -1)t} |11\rangle = e^{-it} |11\rangle \end{aligned}$$

Por lo que se obtiene un factor de fase con un cambio de signo que depende de la paridad de los qubits, por lo que se generaliza como

$$e^{-iZ \otimes Zt} |ab\rangle = e^{-i(-1)^{a \oplus b} t} |ab\rangle \quad (17)$$

La implementación en un circuito se produce como:



En 2, el estado del circuito es:

$$|a\rangle |a \oplus b\rangle$$

En 3 el estado es:

$$|a\rangle \left| e^{-i(-1)^{a \oplus b} t} a \oplus b \right\rangle$$

En 4 Se logra la operación buscada para la ecuación (17)

$$\begin{aligned} |a\rangle \left| e^{-i(-1)^{a \oplus b} t} a \oplus b \oplus a \right\rangle \\ = e^{-i(-1)^{a \oplus b} t} |a\rangle |b\rangle \end{aligned}$$

Para escribir en el circuito el operador *Mixer* B de la ecuación (5) de forma $e^{-i\beta B} = e^{-i\beta \sum_j X_j}$, se analiza término por término.

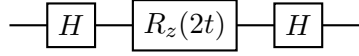
Para escribirlos en términos de la puerta Z , se considera la siguiente serie de Taylor:

$$\begin{aligned} e^{-iZt} &= \sum_{j=0}^{\infty} \frac{(-iZt)^j}{j!} \\ &= I - iZt - \frac{Z^2t^2}{2!} + \dots \end{aligned}$$

Utilizando las compuertas Hadamard H se obtiene:

$$\begin{aligned} He^{-iZt}H &= H \left(I - iZt - \frac{Z^2t^2}{2!} + \dots \right) H \\ &= I - iHZHt - \frac{HZHHZHt^2}{2!} + \dots \\ &= I - iXt - \frac{X^2t^2}{2!} + \dots = e^{-iXt} \end{aligned}$$

Por lo tanto este operador queda implementado en el circuito como



3.2 Simulación de un Computador Cuántico

Para implementar el algoritmo para el problema del viajero, utilizamos el paquete qiskit versión 0.25.0 en Python, en una MacBook Pro Intel Core i5 con 8 GB de RAM 2133 MHz LPDDR3. En una computadora de estas características se puede implementar una cantidad muy limitada de qubits. Como el problema del viajero requiere una cantidad N^2 de qubits para N ciudades, no es posible estudiar problemas complejos de más de seis ciudades.

4 Resultados

El algoritmo QAOA fue implementado para el problema del viajero, considerando un problema con distancias y conexiones aleatorias. Debido a las limitaciones en la implementación cuántica a la que se logró acceder, se consideró el problema para cantidades de ciudades para las que se cuenta con soluciones clásicas, por lo que se verificaron los resultados comparando con resultados obtenidos de forma exacta.

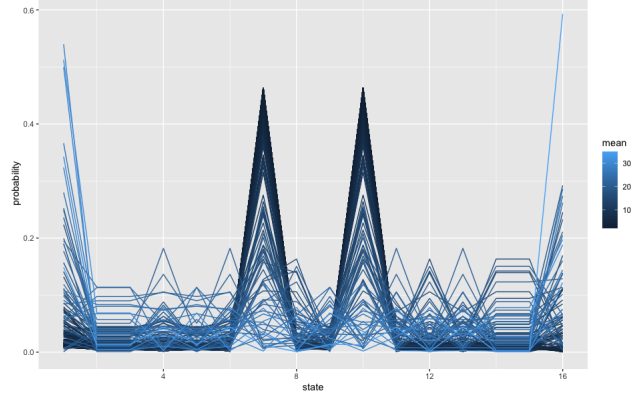


Figura 1: Se muestra los histogramas obtenidos para cada una de las 247 iteraciones de minimización realizadas por el optimizador clásico hasta converger en el valor esperado. Se puede observar la acumulación de probabilidad en los estado solución en medida que la minimización converge

4.1 Implementación e histogramas

En la Figura 1 se puede observar los histogramas obtenidos al aplicar el QAOA a un problema de cuatro ciudades con profundidad $p = 6$ y distintos valores esperados. El optimizador clásico necesito 247 iteraciones para converger en valor esperado. Se puede notar como el peso de las probabilidades se va acumulando en las soluciones buscadas, a medida que el valor esperado converge. Este criterio de minimización de valor esperado es el que utiliza el optimizador clásico (en este caso se utilizó *minimize* del paquete *scipy* de *Python*). Para este problema hay varias soluciones cíclicas equivalentes ya que no se tiene en cuenta de qué ciudad partimos.

En la Figura 2 se puede observar los histogramas obtenidos del muestro de los estados considerando diez profundidades p secuenciales del uno al diez, para un mismo problema. Los parámetros γ y β utilizados en cada distribución son aquellos a los cuales convergió el optimizador en cada caso. Se puede notar como el peso de las probabilidades se va acumulando en las soluciones buscadas. a medida que aumenta la profundidad del algoritmo, el valor esperado es menor, y por lo tanto el espacio de estados variacional Ω_i $i \in \{1, \dots, 10\}$ puede incluir estados mas cercanos a la solu-

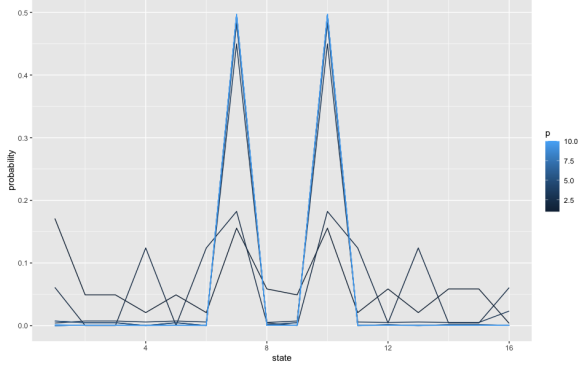


Figura 2: Se muestra los histogramas obtenidos para cada una de las 10 instancias de minimización del valor esperado de un mismo problema con profundidad p creciente del uno al diez. Se puede observar como los estados solución aumentan en probabilidad con mayor profundidad p del algoritmo.

ción.

Observamos que los estados que se miden con mayor probabilidad coinciden con las soluciones exactas con las que se cuenta.

4.2 Convergencias Uniformes en las distribuciones de probabilidad

A partir del muestreo de 39 problemas distintos, en la Figura 3 se muestra como la definición de convergencia uniforme detallada en el Anexo se cumple cuando, para profundidad p fija se minimiza en base del valor esperado. De una distribución inicial de los estados, se converge a la mejor distribución posible dado ese p . El espacio de estados variacional Ω_p generado por un circuito *Ansatz* de profundidad p produce una distribución de probabilidad $F_{\gamma,\beta}$ para cada uno de los posibles estados de Ω_p . Si ordenamos cada una de estas distribuciones por su valor esperado (en orden decreciente) obtenemos una sucesión f_n de distribuciones. Desde la práctica se aprecia que la sucesión converge uniformemente a la mejor distribución en Ω_p .

En la Figura 4 se muestra como, desde la práctica, también existe una convergencia uniforme al variar la profundidad p para un mismo problema. En este caso consideramos la sucesión f_n como una secuencia de funcio-

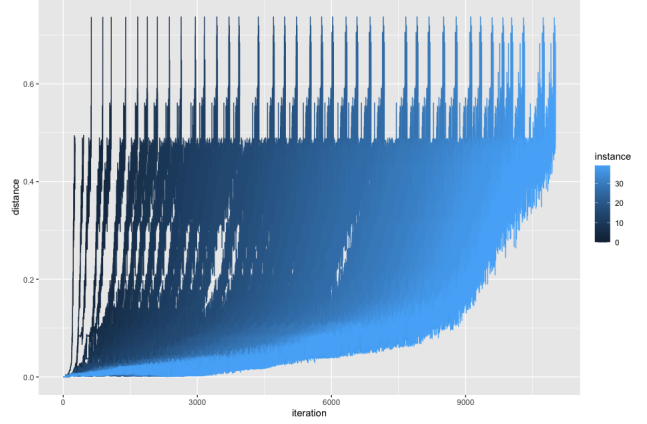


Figura 3: Se muestra las distancias con la distribución mínima en cada iteración de minimización de 39 instancias de problemas distintos ejecutados. En el eje de abscisas se mide el número de iteración del optimizador y en el eje de ordenadas se muestra la distancia con la función f de distribución mínima que alcanzó el optimizador a la cual converge la sucesión f_n . Para cada una de las instancias ejecutadas se cumple la definición de convergencia uniforme $f_n \xrightarrow{\text{c.u.}} f$.

nes de distribución generada por las distribuciones de mínimo valor esperado que alcanzó el optimizador para cada sucesiva profundidad p . Los espacios variacionales generados en cada paso de la sucesión incluyen a los generados en pasos previos. Observando la ecuación (5) podemos definir a medida que aumenta la profundidad, los espacios Ω_i se contienen crecientemente, de manera que $\Omega_1 \subset \Omega_2 \dots \subset \Omega_i \dots \subset \Omega_p$. Como en la Figura 3 se mostró que para cualquier Ω_p la convergencia es uniforme, intuitivamente es posible interpretar que la convergencia uniforme en Ω_i , contiene a la convergencia en Ω_{i-1} , y así para sucesivos valores decrecientes de profundidad. Entonces, los mínimos de sucesivos Ω_i tienen una convergencia uniforme para valores decrecientes de valor esperado.

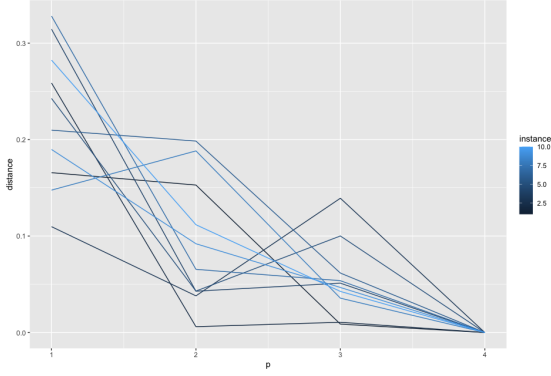


Figura 4: Se muestra las distancias con la distribución mínima alcanzada con un $p = 4$ en diez instancias y variando la profundidad p del uno al cuatro. En el eje de abscisas se mide la profundidad p y en el eje de ordenadas la distancia con la función de distribución mínima alcanzada con un $p = 4$. Para cada una de las instancias ejecutadas se cumple la definición de convergencia uniforme.

4.3 Interpretación estadística del QAOA mediante el mínimo en la muestra y las convergencias uniformes

En la subsección 4.2 se hace una demostración de posibles convergencias uniformes en las distribuciones de probabilidad para los estados cuánticos. Si asumimos que esas convergencias existen matemáticamente y están demostradas, es un punto importante porque se puede utilizar para darle una interpretación estadística al algoritmo QAOA.

A partir de estas convergencias, se encuentra la siguiente explicación del algoritmo: mediante la minimización de los parámetros γ y β y el aumento de la profundidad p del circuito *Ansatz*, aumenta la probabilidad de obtener el estado que minimiza el costo al realizar mediciones. De este modo tenemos un algoritmo de muestreo, donde la muestra se va restringiendo a los estados más favorables en la medida en que se mejora la elección de parámetros o se aumenta la profundidad. El objetivo es garantizar que ese muestreo incluya la solución deseada.

Las convergencias uniformes formalizan el criterio de aproximación del algoritmo, ya que

cualquiera sea la profundidad p alcanzada en la práctica, e independientemente del valor esperado mínimo alcanzado, se asegura que la distribución reflejará cada vez más la solución del problema. Por otro lado formalizan el criterio de optimización mediante el valor esperado, ya que gracias a las convergencias uniformes un estado *Ansatz* con menor valor esperado que otro es una mejor opción para el muestreo, ya que nos asegura que se favorecieron en probabilidad los estados cercanos a la solución deseada.

5 Conclusiones

En este artículo, se explicó en detalle el algoritmo QAOA, su origen y su implementación. Se logró obtener resultados coherentes con los resultados clásicos para el problema del viajero para problemas aleatorios.

La observación de los resultados permitió proponer interpretaciones de su funcionamiento, que amplían lo propuesto en el artículo original. También se interpretó el algoritmo QAOA en el contexto de los algoritmos variacionales. Se presenta la posible existencia de convergencias uniformes en el procedimiento del algoritmo.

Dejamos abierta la puerta para, a futuro, encontrar la demostración matemática formal de las convergencias, y su generalización para otros problemas de optimización.

Referencias

- [1] John Preskill. «Quantum Computing in the NISQ era and beyond». En: *Quantum* 2 (ago. de 2018), pág. 79. ISSN: 2521-327X. DOI: 10.22331/q-2018-08-06-79. URL: <http://dx.doi.org/10.22331/q-2018-08-06-79>.
- [2] Quantum Computing. *Quantum Approximate Optimization Algorithms (Peter Shor, ISCA 2018)*. Youtube. 2018. URL: <https://www.youtube.com/watch?v=HHIWUi3GmdM>.

- [3] Edward Farhi, Jeffrey Goldstone y Sam Gutmann. *A Quantum Approximate Optimization Algorithm*. 2014. arXiv: 1411.4028 [quant-ph].
- [4] Stuart Hadfield. *On the representation of Boolean and real functions as Hamiltonians for quantum computing*. 2018. arXiv: 1804.09130 [quant-ph].
- [5] Andrew Lucas. «Ising formulations of many NP problems». En: *Frontiers in Physics* 2 (2014). ISSN: 2296-424X. DOI: 10.3389/fphy.2014.00005. URL: <http://dx.doi.org/10.3389/fphy.2014.00005>.
- [6] *Supremacía Cuántica*. Wikipedia. URL: <https://es.wikipedia.org/wiki/Supremaciacuantica>.
- [7] The Unconventional Computing Lab. *Curso Algoritmos Variacionales Cuánticos*. URL: <https://www.youtube.com/channel/UCxafWublayjixDsAAGM2tZA>.
- [8] Qiskit. *Solving combinatorial optimization problems using QAOA*. URL: <https://qiskit.org/textbook/ch-applications/qaoa.html>.
- [9] Dorit Aharonov y col. «Adiabatic Quantum Computation is Equivalent to Standard Quantum Computation». En: (2004). DOI: 10.48550/ARXIV.QUANT-PH/0405098. URL: <https://arxiv.org/abs/quant-ph/0405098>.
- [10] Tameem Albash y Daniel A. Lidar. «Adiabatic quantum computation». En: *Reviews of Modern Physics* 90.1 (ene. de 2018). DOI: 10.1103/RevModPhys.90.015002. URL: <https://doi.org/10.1103/RevModPhys.90.015002>.
- [11] Jaimie Stephens, Ciaran Ryan-Anderson y Ojas Parekh. *Quantum Approximate Optimization Algorithm (QAOA) on Constrained Optimization Problems*. URL: <https://www.osti.gov/servlets/purl/1577041>.
- [12] *Solving problems with quantum samplers*. URL: https://docs.dwavesys.com/docs/latest/c_gs_3.html#solving-problems-with-quantum-samplers.
- [13] Ruslan Shaydulin. *A tutorial on Quantum Approximate Optimization Algorithm*. Youtube. 2020. URL: <https://www.youtube.com/watch?v=AOKM9BkweVU&t=1849s>.

Anexo

Convergencia de Funciones y Convergencia Uniforme

Dada dos funciones que actúan en un mismo espacio, la idea es primero definir la cercanía de dos funciones. Si se tiene un espacio vectorial V se puede definir una norma $\|\cdot\|$ asociada al espacio.

$$V \implies \|x\|/x \in V$$

La norma cumple con condiciones $\|\vec{x}\| \geq 0$ y la norma $\|\vec{x}\| = 0$ si y solo si $\vec{x} = \vec{0}$. Generalizando $\|\vec{x} - \vec{y}\|$, en cualquier conjunto A , dado dos elementos $\vec{x}, \vec{y} \in A$, se puede definir una distancia $d(\vec{x}, \vec{y})$. Si d es una función tal que:

$$d : A \times A \implies \mathbb{R}$$

$$d(\vec{x}, \vec{y}) > 0, \vec{x} \neq \vec{y}$$

$$d(\vec{x}, \vec{x}) = 0$$

$$d(\vec{x}, \vec{z}) \leq d(\vec{x}, \vec{y}) + d(\vec{x}, \vec{z})$$

Dada dos funciones f y g que pertenecen a un conjunto de funciones A se puede definir

$d(f, g) = \sup\{\|f(x) - g(x)\| : x \in M\}$, donde M es el dominio de interés de las funciones.

En una sucesión de funciones dada f_n que se acerca en la medida de distancia anterior a una función f se dice que f_n converge uniformemente a f , es decir

$$d(f_n, f) \rightarrow 0 \implies f_n \xrightarrow{\text{c.u.}} f$$

Desde una perspectiva mas formal se define la convergencia uniforme entre dos funciones como

Dado

$$\epsilon > 0, \exists n_0 = n_0(\epsilon) > 0$$

$$\|f_n(x) - f(x)\| < \epsilon, \forall x \in M$$

si

$$n \geq n_0$$